An Encryption Based Technique for Invisible Digital Watermarking

Samir Kumar Bandyopadhyay
Dept. of Computer Sc. & Engg,
University of Calcutta
92 A.P.C. Road, Kolkata – 700009,
India

Tuhin Utsab Paul
Dept. of Computer Sc. & Engg,
University of Calcutta,
92 A.P.C. Road, Kolkata-700009,
India

Avishek Raychoudhury
Dept. of Computer Sc. & Engg,
University of Calcutta
92 A.P.C. Road, Kolkata-700009,
India

ABSTRACT
In this paper, a technique for hiding the data of images has been proposed. The proposed method is used to hide an image file entirely with in another image file keeping two considerations in mind, which are Size and Degree of Security. At the source, the image that is to be hidden (target image) is encoded within another image (cover image). Firstly, the cover image and the target image can be of any size, which will be adjusted by our resize function thereby, removing the size constraint. Secondly, for the security of transmission over network, only the final encrypted image i.e. cover image and target image is sent over the network. This image is easily decoded at the receiver end. Results are encouraging from practical point of view.

Keywords
Data, image, Hiding, Security, Encryption

1. INTRODUCTION
Data hiding is defined as the process by which a message or image is imperceptibly embedded into a host or cover to get a composite signal. Steganography is the art and science of writing hidden messages in such a way that no one apart from the sender and intended recipient even realizes there is a hidden message. Generally, in steganography, the actual information is not maintained in its original format and thereby it is converted into an alternative equivalent multimedia file like image, video or audio, which in turn is being hidden within another object. This apparent message (known as stego-object in usual terms) is sent through the network to the recipient, where the actual message is separated from it.

In this paper, we have considered some important features of data hiding. Our first consideration is that of embedding information into image, which could survive attacks on the network. Next, a hybrid digital embedding technique is proposed for hiding an image into another image in such a way that the quality of the recovered image improves significantly. Also to make the proposed scheme to run free of size constraints, we have introduced the concept of Resizing before doing the actual hiding.

Lastly, in most of the algorithms designed based on the principle of Steganography, requires the sending original cover image along with the encoded cover image to the receiver. This approach makes the designed algorithm weaker as it conveys some idea of data hiding to the sender. But our method only the encoded image will be sent to the receiver.

To the best of our knowledge, this work is specifically focused on protection of any information, which is in the form of image. The design of this technique is based on extensive analytical as well as experimental modeling of the data-hiding process.

2. PREVIOUS WORKS
The majority of today's steganographic system uses images as cover media because people often transmit digital pictures over email and other Internet communication. Several methods exist to employ the concept of Steganography as well as plenty algorithms have been proposed in this regard. To propose our approach, we have concentrated on some techniques and methods, which are described below.

In the field of image security, Miroslav Dobsicek [1] has developed an interesting application of steganography where the content is encrypted with one key and can be decrypted with several other keys, the relative entropy between encrypt and one specific decrypt key.

Corresponds to the amount of information Yusuk Lim, Changsheng Xu and David Dagan Feng, 2001, described the web based authentication system consists of two parts: one is a watermark embedding system and the other is authentication system. In case of watermark embedding system, it is installed in the server as application software that any authorized user, who has access to server, can generate watermarked image. The distribution can use any kind of network transmission such as FTP, email etc. Once image is distributed to externally, client can access to authentication web page to get verification of image [2].

Min Wu and Bede Liu, June 2003, proposed [3] a new method to embed data in binary images, including scanned text, figures, and signatures. The method manipulates "flappable" pixels to enforce specific block based relationship in order to embed a significant amount of data without causing noticeable artifacts. They have applied Shuffling before embedding to equalize the uneven embedding capacity from region to region. The hidden data can then be extracted without using the original image and can also be accurately extracted after high quality printing and scanning with the help of a few registration marks.

Rehab H. Alwan, Fadhil J. Kadhim, and Ahmad T. Al- Taani, 2005, have explained a method with three main steps. First, the edge of the image is detected using Sobel mask filters. Second, the least significant bit LSB of each pixel is used. Finally, a gray level connectivity is applied using a fuzzy approach and the

ASCII code is used for information hiding. The prior bit of the LSB represents the edged image after gray level connectivity, and the remaining six bits represent the original image with very little difference in contrast. The given method embeds three images in one image and includes, as a special case of data embedding, information hiding, identifying and authenticating text embedded within the digital images [4].

In 2007, Nameer N. EL-Emam proposed an algorithmic approach to obtain data security using LSB insertion steganographic method. In this approach, high security layers have been proposed through three layers to make it difficult to break through the encryption of the input data and confuse steganalysis too [5].

Prof S. K. Bandyopadhyay, Debnath Bhattacharyya, Swarnendu Mukherjee, Debashis Ganguly, Poulami Das in 2008 has proposed a heuristic approach to hide huge amount of data using LSB steganography technique. In their method, they have first encoded the data and afterwards the encoded data is hidden behind a cover image by modifying the least significant bits of each pixel of the cover image. The resultant stego-image was distortion less. Also, they have given much emphasis on space complexity of the data hiding technique [6].

There is also a good method proposed by G. Sahoo and R. K. Tiwari in 2008. Their proposed method works on more than one image using the concept of file hybridization. This particular method implements the cryptographic technique to embed two information files using steganography and due to this reason they have used a stego key for the embedding process [7]. Unfortunately, modifying the cover image changes its statistical properties, so eavesdroppers can detect the distortions in the resulting stego-image's statistical properties. In fact, the embedding of high-entropy data (often due to encryption) changes the histogram of colour frequencies in a predictable way. So, in order to obtain more security in our prescribed method, we have embedded an entire image behind another image of twice the size of target image for a remarkable change in the final image.

3. PROPOSED WORK

Before going into the details of the algorithm proposed here for invisible watermark of the information behind the cover object, it is better to mention about the selection of the images and information which are to be steganographed. Here in this paper, the algorithm is basically implemented over normal bitmap image file, but it should be clarified that the same scheme can be extended to operate over other file formats also. The image file, which is to be hidden, is here referred as Target Image and the image behind which it is to be hidden is termed as Cover Image. The selection of neither the Target Image nor the Cover Image is constrained by any size limit.

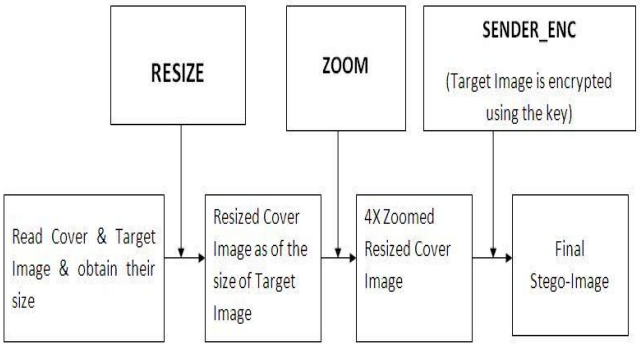
After selecting the pictures, we have to resize the Cover Image with the size of the Target Image using Bi-cubic interpolation technique. The resizing will be done in such a way that after resizing the size of the Cover Image will be equal to the size of the Target Image.

Next the Cover Image is zoomed twice of its original size using row-column duplication scheme ie, say a pixel (x,y) is duplicated in {(x', y'), (x'+1,y'), (x',y'+1), (x'+1,y'+1)}. In next attempt, the entire Target Image will be hidden in the Cover Image starting from the first byte position of the Cover Image.

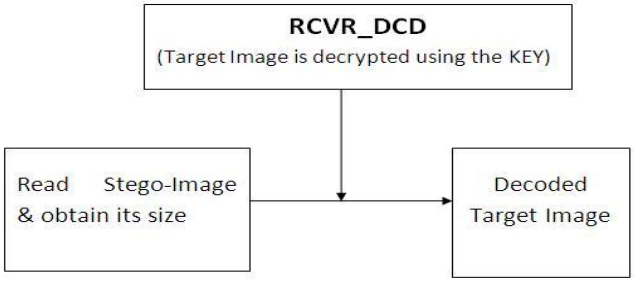
The method for encryption can be personalized, i.e., can be selected according to the user needs. But, the authors specifically suggests this specialized scheme, proposed in this paper, as because here the information are no longer being merged or masked with another and instead of that keeping the pixel values of the Target Image as an encrypted information in the carrier, i.e., the Cover Image is altered to obtain resultant image which is taken as Final Image. Thus no essence of the actual information is retrained in the Final Image, whereas in usual methods of the mostly done bitwise merging; the information belongs in encrypted way directly merged into final object obtained.

The stated Algorithm has got five distinct divisions: a. main function for each of the sender and receiver end which calls next sections; b. to resize the Cover Image in the size of the Target Image. c. to zoom the Cover Image in twice the size of the Target Image d. Encryption; e. Decryption.

SENDER END APPROACH ::



RECIEVER END APPROACH ::



3.1 Algorithms

3.1.1 SNDR_MAIN (Target Image, Cover Image)

This is the main function in our algorithm. This function will also be used in the sender side and will call other modules of our algorithm.

Input: This function will take Target Image and Cover Image as input.

Output: It will output the encoded stego-image.

1. Obtain the size of the Target Image and store it as Target Image Size.
2. Now choose the Cover Image and obtain its size and store it in Cover Image Size.
3. Now call the Resize module with the Target Image Size and Cover Image as arguments.
4. Next call the Zoom function to transform the Cover Image in the size 4 times of that of Target Image using row-column duplication technique.
5. Now call the SNDR_ENC module to hide the Target Image behind the Cover Image with arguments Cover Image, Target Image and thereby obtain the final Image.
6. Send only the final Image over the network to reach the intended destination.

3.1.2 RESIZE (PICTURE, SIZE)

This function is used in the algorithm to resize an image to obtain an image of the desired size from the input image.

Input: This function will take the image, which has to be resized along with the desired image size, which is to be obtained after resizing.

Output: This function outputs an image of desired size.

1. Obtain the width, length of the final image that has to be produced.
2. Use Bi-cubic interpolation technique to resize (zoom/shrink) the image.
3. Return the PICTURE.

3.1.3 ZOOM (PICTURE, SIZE)

This function is used in the algorithm to zoom an image to obtain an image of the double size of the input image using row-column duplication technique.

Input: This function will take the image, which has to be zoomed along with the desired image size, which is to be obtained after zooming.

Output: This function outputs an image of desired size.

1. Obtain the width, length of the final image that has to be produced (say R, C).
2. Create a blank matrix of size R x C.
3. Read each pixel(say r, c) from the input image and copy the value in four pixels of the new matrix {(r', c'), (r'+1,c'), (r',c'+1), (r'+1,c'+1)}
4. Return the PICTURE formed from the new matrix.

3.1.4 SNDR_ENC (PICTURE_1, PICTURE_2)

This function is used in the algorithm to encrypt an image with the help of another image to obtain an encrypted image.

Input: This function will take the cover image (PICTURE_1) [Resized and Zoomed 4x] in which another image will be hidden i.e. the target image (PICTURE_2).

Output: This function will output the stego-image as the final image.

1. Read both the PICTURE_1, PICTURE_2.
2. Read one pixel (r,c) from PICTURE_2 and set r'=2*r and c'=2*c.(val(x,y) is the pixel value of PICTURE_1 at position (x,y)).
3. Divide the pixel value read from PICTURE_2 using the KEY {KEY|KEY belongs to S, where S is a set of all positive integers between 2 to 255} and store the dividend as Div and remainder as Rem.
4. Let val is a variable. Set, value=val(r',c'-1)+Div.
5. If value>255 then store val(r',c'-1)=value-255.
   Else val(r',c'-1)=val.
6. Let val1 is a variable. set, val1=val(r'-1,c')+Rem.
7. If val1>255 then store val(r'-1,c')=val1-255.
8. Repeat step 2 to 7 until all the pixels of PICTURE_2 are encoded in the PICTURE_1.
9. Insert the key in the free byte of the image header.
10. Return the PICTURE_1.

3.1.5 RCVR_MAIN (STEGOIMAGE)

This is the main function in our algorithm. This function will be used in the receiver side and will call other modules of our algorithm.

Input: This function will take StegoImage as input.

Output: It will output the decoded Image.

1. Obtain the size of the StegoImage and store it as StegoImageSize say r',c'.
2. Now call the RCVR_DCD module to decode the StegoImage to get the hidden image with arguments StegoImage .
3. The final Image is retrived from the StegoImage.

3.1.6 RCVR_DCD (PICTURE_1)

This function is used in the algorithm to decrypt an image with the help of a key to obtain the final image.

Input: This function will take the StegoImage (PICTURE_1) in which another image is hidden.

Output: This function will output the TargetImage as the final image.

1. Read the PICTURE_1 and retrieve the Key from its header and define a matrix of size (r'/2,c'/2) say (m,n). (val(x,y) is the pixel value of PICTURE_1 at position (x,y)).
2. set r=1,c=1,i=1,j=1;
3. Read pixel values:val(r,c),val(r+1,c),val(r,c+1).
4. if (val(r,c)>val(r+1,c) and val(r,c)>val(r,c+1))then
   value=((255+val(r+1,c)-val(r,c))*Key)+(255+val(r,c+1)-val(r,c))
   else
   if(val(r,c)<=val(r+1,c) and val(r,c)>val(r,c+1))then
   value=((val(r+1,c)-val(r,c))*key)+((255+val(r,c+1)-val(r,c))
   else
   if(val(r,c)>val(r+1,c) and val(r,c)<=val(r,c+1))then
   value=((255+val(r+1,c)-val(r,c))*Key)+(val(r,c+1)-val(r,c))
   else
   value=((val(r+1,c)-val(r,c))*key)+(val(r,c+1)-val(r,c))
5. Store 'value' in position (i,j) of the new matrix.
6. Increment (i,j) to point the next cell of the new matrix in row major order.

7. increment r,c to read the next quadrant of the PICTURE_1.
8. Repeat step 3 to 7 until the PICTURE_1 is exhausted.
9. Return the new matrix as the FinalImage.

4. TEST RESULT

4.1 Complexity analysis of the stated algorithm

In the sender end, the resize and Zoom function both scan through the image in row major order. So for image dimension mXn the time complexity is of order O(mn). The encryption algorithm performs parallel scan on both the image in row wise order thereby giving the time complexity of the order O(mn). Thus in the sender end the order of time complexity becomes O(mn).

In case of space complexity at the sender end, for coverimage dimension mXn, after the zooming operation, the size becomes 2mX2n.So,for storing this zoomed image, space required=2m*2n=4mn,which is O(mn).The target image is then encrypted and stored in this zoomed image, which does not alter the space complexity.

In the receiver end the decryption algorithm performs image scan in row wise order and generate the target image. Thus the time complexity order becomes O(mn).

In case of space complexity at the receiver end, if the received image size is mXn,then the FinalImage is (m/2Xn/2). So, for storing the FinalImage space required=(m/2)*(n/2)=mn/4,which is O(mn).

4.2 Test Results

SENDER END ::

Figure 1 :CoverImage (300x280)

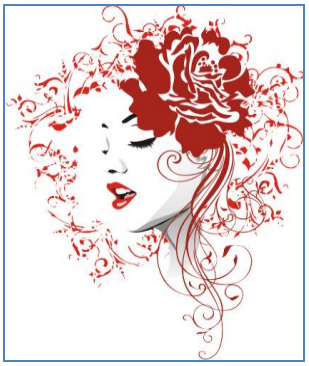


Figure 2:TargetImage(513x420)

Figure 3:StegoImage (1026x840)

RECIEVER END ::

Figure 4:FinalImage (513x420)

COMPARATIVE HISTOGRAMS ::

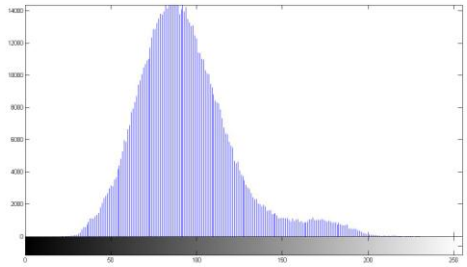


Figure 5 : Histogram of CoverImage before encryption

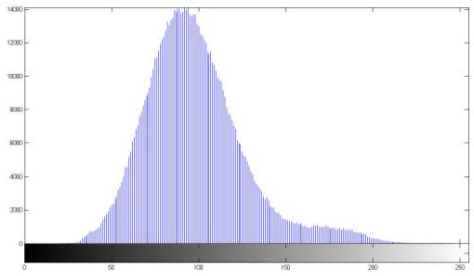


Figure 6 : Histogram of StegoImage after encryption

Figure 1: Original Cover Image considered before encryption.

Figure 2: Original Target Image considered before encryption.

Figure 3: Stego Image obtained after encryption at sender end.

Figure 4: Target Image retrieved after decryption of received Stego Image.

Figure 5: Histogram of CoverImage before encryption.

Figure 6: Histogram of StegoImage after encryption.

5. CONCLUSION

From the 2 histograms given above of the zoomed cover image(before and after encryption),we can clearly see that the difference among the 2 image histograms is nominal. So, for any Statistical Steganalysis tool or any other intruder it is nearly impossible to trace the hidden image inside the cover image.

The main advantage of our algorithm is that the final image can be derived only from the StegoImage. The original cover image is not needed for decoding the stego image. This provides less network transmission overhead as well as less scope of suspicion for the network intruder. Moreover this algorithm is free from size constrains i.e. it performs well on any size of the cover image or target image. The fourth pixel value of every quadrant of the stego image is free and using LSB modification can use it for transmission of additional data or DCT based method or any other method.

Apart from storing the val in SNDR_ENC directly by summing with the pixel value it can be encoded using any encoding technique and then added thereby adding extra security to the hiding technique.

6. REFERENCES

[1] Dobsicek, M., Extended steganographic system. In: 8th Intl. Student Conf. on Electrical Engineering, FEE CTU 2004, Poster 04.

[2] 2. Yusuk Lim, Changsheng Xu and David Dagan Feng, "Web based Image Authentication Using Invisible Fragile Watermark", 2001, Pan-Sydney Area Workshop on Visual Information Processing (VIP2001), Sydney, Australia, Page(s): 31 - 34

[3] 3. Min Wu, Member, IEEE, and Bede Liu, Fellow, IEEE, "Data Hiding in Binary Image for Authentication and Annotation", IEEE Trans. Image Processing, volume 6, Issue 4, Aug. 2004 Page(s): 528 - 538

[4] 4. Rehab H. Alwan, Fadhil J. Kadhim, and Ahmad T. Al-Taani, "Data Embedding Based on Better Use of Bits in Image Pixels", International Journal of Signal Processing Vol 2, No. 2, 2005, Page(s): 104 - 107

[5] 5. Nameer N. EL-Emam "Hiding a large amount of data with high security using steganography algorithm", Journal of Computer Science. April 2007, Page(s): 223 – 232

[6] S.K.Bandyopadhyay, Debnath Bhattacharyya, Swarnendu Mukherjee, Debashis Ganguly, PoulumiDas, "A Secure Scheme for Image Transformation", August 2008, IEEE SNPD, Page(s): 490 – 493

[7] G. Sahoo, R. K. Tiwari, "Designing an Embedded Algorithm for Data Hiding using Steganographic

[8] Technique by File Hybridization", January 2008, IJCSNS, Vol. 8, No. 1,Page(s): 228 – 2336. S.K.Bandyopadhyay, Debnath Bhattacharyya, Swarnendu Mukherjee, Debashis Ganguly, Poulumi Das, "A Secure technique for Image data hiding".